```
greeting: 'r' 's' 'm' 't' 'p' ' ' role ' ' DOMAIN '\n';
role:  't' I 'r';


envelope: 'i' '\n' I 'i' ADDRESS ' ' numb ' ' notify '\n' 'm' NON_NL '\n'
        optext 's' ADDRESS '\n' rcpts '\n';
notify: 'n' I 'z' I 'b' I 'c';
optext :  I LETTER NON_NL '\n' optext;
rcpts: 't' ADDRESS '\n' I 't'ADDRESS '\n' rcpts;



envstatus:    optextrepl rcpstats sendstatus '\n';
optextrepl:   I  optext '\n';
rcpstats:     rcpstat '\n' I rcpstat '\n' rcpstats ;
rcpstat:      msgstat I rcpreject NON_NL ;
rcpreject:    'n' I 'f';



sendstatus:   'Y' I 'R' numb ' ' NON_NL I sendreject NON_NL ;
sendreject:   'N' I 'F';



message:    fragment I  fragment message ;
fragment:   numb '\n' ANYBYTES '.' ;
msgstatus:  msgstat '\n';
msgstat:    'y' I 'r' numb ' ' NON_NL ;
numb:   digit I digit numb ;
digit:    '0' I '1' I '2' I '3' I '4' I '5' I '6' I '7' I '8' I '9';
```
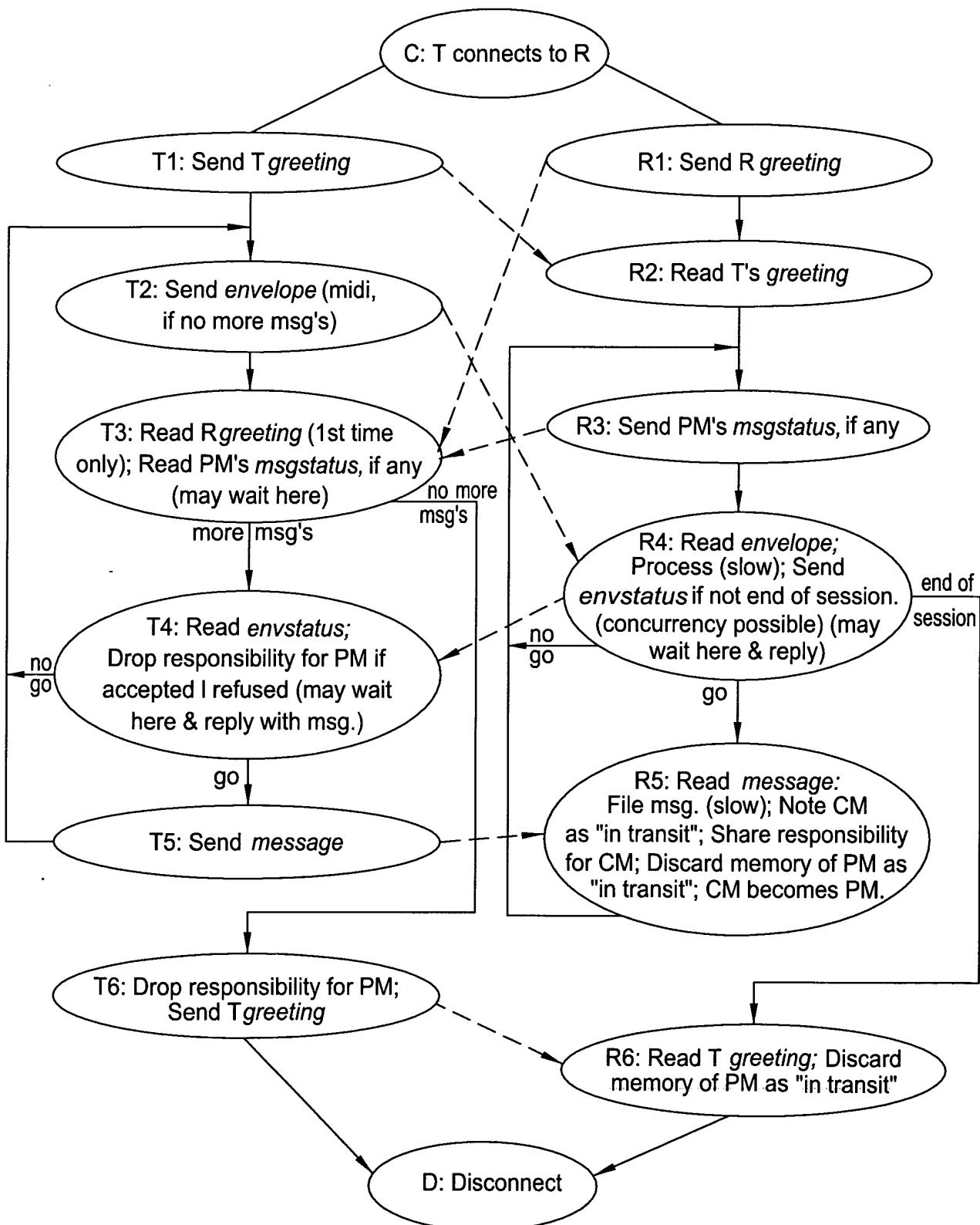
FIG. 1

OIPE JC124
JUL 29 2005
PATENT & TRADEMARK OFFICE

C: T connects to R

T1: Send T *greeting*

R1: Send R *greeting*

T2: Send *envelope* (midi, if no more msg's)

R2: Read T's *greeting*

T3: Read R *greeting* (1st time only); Read PM's *msgstatus,* if any (may wait here)

R3: Send PM's *msgstatus,* if any

no more msg's

more msg's

R4: Read *envelope;* Process (slow); Send *envstatus* if not end of session. (concurrency possible) (may wait here & reply)

end of session

no go

T4: Read *envstatus;* Drop responsibility for PM if accepted I refused (may wait here & reply with msg.)

no go

go

R5: Read *message:* File msg. (slow); Note CM as "in transit"; Share responsibility for CM; Discard memory of PM as "in transit"; CM becomes PM.

go

T5: Send *message*

T6: Drop responsibility for PM; Send T *greeting*

R6: Read T *greeting;* Discard memory of PM as "in transit"

D: Disconnect

FIG. 2